**Date Submitted: 10/21/2019**

**Task 00: Execute provided code**

**Provided code with comprehensive comments:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> //new include uses the sinf() function

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/fpu.h" //new include for floating point unit support

**#include** "driverlib/sysctl.h"

**#include** "driverlib/rom.h"

**#include** "driverlib/rom\_map.h"

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846 //pie value

**#endif**

**#define** SERIES\_LENGTH 100 //depth of the data buffer

**float** gSeriesData[SERIES\_LENGTH]; //an array of floats SERIES\_LENGTH long

int32\_t i32DataCount = 0; //counter variable for the sine computation loop

**int** **main**(**void**)

{

**float** fRadians; //variable that calculates the value of sine

//Lazy stacking will avoid an increase of interrupt latency by skipping the stacking of floating point registers.

ROM\_FPULazyStackingEnable(); //enables lazy stacking of floating point registers s0-s15 when an interrupt is handled.

ROM\_FPUEnable(); //turns on floating point unit

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //50MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); //full sine wave is 2pi radians. 2pi/series length gives radians

**while**(i32DataCount < SERIES\_LENGTH) //calculates sine value for each of 100 values

{

gSeriesData[i32DataCount] = **sinf**(fRadians \* i32DataCount); //sine calculation

i32DataCount++; //increment counter for while loop

}

**while**(1) //run infinitely

{

}

}A screenshot of a social media post

Description automatically generated

A picture containing screenshot

Description automatically generated

**------------------------------------------------------------------------------------**

**Task 01:**

Youtube Link:

**Modified Code:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> //new include uses the sinf() function

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/fpu.h" //new include for floating point unit support

**#include** "driverlib/sysctl.h"

**#include** "driverlib/rom.h"

**#include** "driverlib/rom\_map.h"

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846 //pie value

**#endif**

**#define** SERIES\_LENGTH 1000 //buffer length of 1000 sample points required by task 1

**float** gSeriesData[SERIES\_LENGTH]; //an array of floats SERIES\_LENGTH long

int32\_t i32DataCount = 0; //counter for the while loop to operate 1000 times

**int** **main**(**void**)

{

**float** fRadians\_one; //variable that calculates the value of sine

**float** fRadians\_two; //variable that calculates the value of cosine

//Lazy stacking will avoid an increase of interrupt latency by skipping the stacking of floating point registers.

ROM\_FPULazyStackingEnable(); //enables lazy stacking of floating point registers s0-s15 when an interrupt is handled.

ROM\_FPUEnable(); //turns on floating point unit

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //50MHz

fRadians\_one = ((2 \* M\_PI \* 50) / SERIES\_LENGTH); //full sine wave is 2pi radians. 2pi/series length gives radians

fRadians\_two = ((2 \* M\_PI \* 200) / SERIES\_LENGTH); //full sine wave is 2pi radians. 2pi/series length gives radians

**while**(i32DataCount < SERIES\_LENGTH) //calculates sine value for each of 100 values

{

//calculate the equation given in task 1, must use the sine loop calculation (fRadians \* i32DataCount) formula for "t"

//1.5 + 1.0\*sin(2pi50t) + 0.5\*cos(2pi200t)

//store the value in a floating point array to display the graph later

gSeriesData[i32DataCount] = 1.5 + (1.0 \* **sinf**(fRadians\_one \* i32DataCount)) + (0.5 \* **cosf**(fRadians\_two \* i32DataCount));

i32DataCount++; //increment counter for while loop

}

**while**(1) //run infinitely

{

}

}

**A picture containing water

Description automatically generated**